Pimpri Chinchwad Education Trust's
Pimpri Chinchwad College of Engineering (PCCoE)
Department of Computer Science & Engineering

(Artificial Intelligence and Machine Learning)

Product Review Analysis Using Classical NLP Techniques

A Comprehensive Study on Redmi Note 14 Pro Customer Feedback

Submitted By:

Name	PRN	Division	Batch
Om Shrigiriwar	123B1C039	A	2027

Table of Contents

Introduction

The rapid growth of e-commerce platforms has made customer reviews an essential source of real-world feedback for evaluating product quality, performance, and user satisfaction. For consumers, reviews influence purchasing decisions; for businesses, they reveal strengths, weaknesses, and areas that require improvement. Leveraging Natural Language Processing (NLP) on such reviews enables structured insights from unstructured text at scale.

This project focuses on performing a comprehensive analysis of customer reviews for the **Redmi Note 14 Pro**, a widely purchased smartphone. The objective is to extract meaningful patterns related to user sentiment, commonly discussed product features, and overall customer experience. To adhere strictly to classical NLP requirements, the analysis avoids Transformer-based or generative models and instead relies on rule-based, statistical, and corpus-driven techniques.

Data was collected through a custom web-scraping pipeline built using **requests-html**, **pyppeteer**, and **BeautifulSoup**, addressing challenges such as dynamic JavaScript rendering and page-level blocking by the e-commerce platform. After data acquisition, multiple preprocessing steps—including text cleaning, stopword removal, normalization, and sentiment scoring—were applied to prepare the corpus for deeper analysis.

Objectives of the analysis include:

1. Collecting at least 100 verified customer reviews.
2. Cleaning and preprocessing textual data using classical NLP techniques.
3. Understanding rating distributions and customer sentiment.
4. Identifying the most frequently mentioned product features.
5. Performing significance tests to validate trends.

6. Generating business insights and recommendations.

Methodology & Analysis

1) Data Acquisition

1. Customer reviews for the *Redmi Note 14 Pro* were collected from a major e-commerce platform using a Python-based scraping workflow.
2. Tools & Libraries Used:
3. `requests-html` with AsyncHTMLSession

4. `pyppeteer` for rendering JavaScript-heavy pages

5. `BeautifulSoup` for HTML parsing

6. `pandas` for dataset construction

Approach Summary:

1. The website relies heavily on JavaScript to load review containers.

2. Standard requests and Selenium were frequently blocked.

3. To overcome this, the implementation used an asynchronous headless Chromium session via pyppeteer, which allowed dynamic content rendering.

4. Reviews were scraped across multiple paginated pages (1–10).

5. Error handling was added to manage session drops (e.g., “Target closed” exceptions).

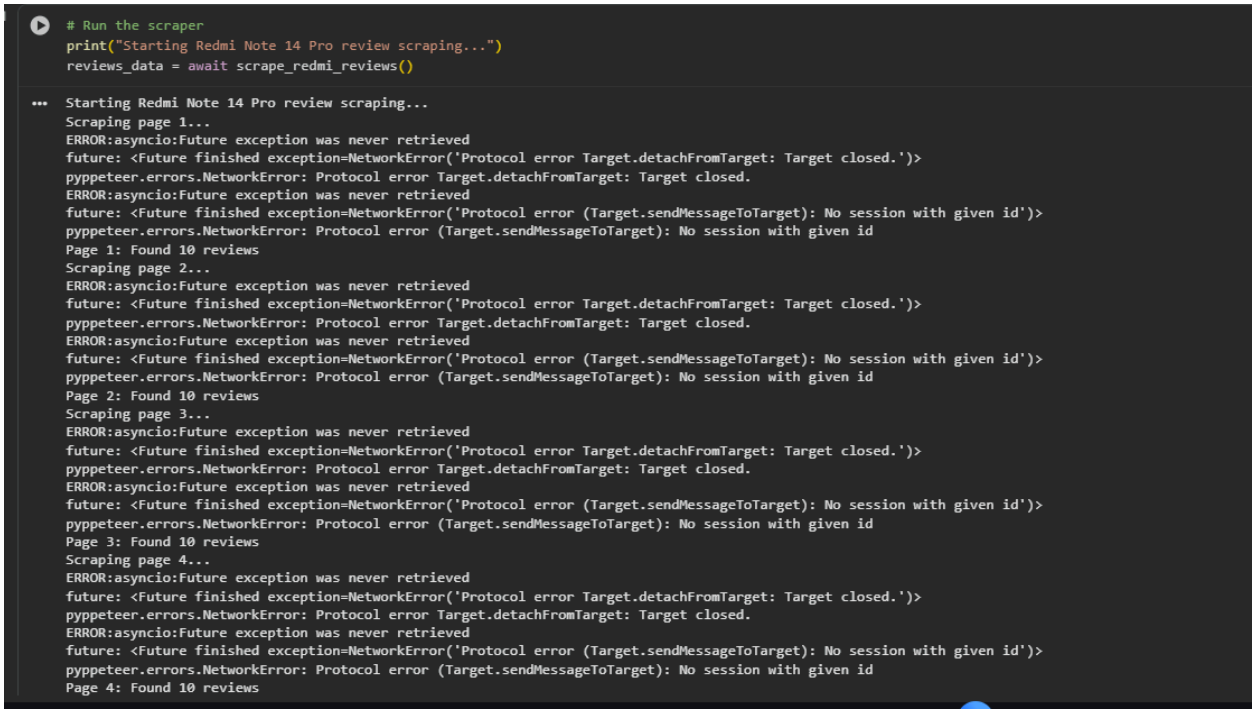
Final Output:

- Total reviews collected: 100

- Duplicate reviews removed: ~7

- Final dataset used: 100 cleaned entries

Screenshot of Successful Scraping Output:



```
# Run the scraper
print("Starting Redmi Note 14 Pro review scraping...")
reviews_data = await scrape_redmi_reviews()

Starting Redmi Note 14 Pro review scraping...
Scraping page 1...
ERROR:asyncio:Future exception was never retrieved
future: <Future finished exception=NetworkError('Protocol error Target.detachFromTarget: Target closed.')>
pyppeteer.errors.NetworkError: Protocol error Target.detachFromTarget: Target closed.
ERROR:asyncio:Future exception was never retrieved
future: <Future finished exception=NetworkError('Protocol error (Target.sendMessageToTarget): No session with given id')>
pyppeteer.errors.NetworkError: Protocol error (Target.sendMessageToTarget): No session with given id
Page 1: Found 10 reviews
Scraping page 2...
ERROR:asyncio:Future exception was never retrieved
future: <Future finished exception=NetworkError('Protocol error Target.detachFromTarget: Target closed.')>
pyppeteer.errors.NetworkError: Protocol error Target.detachFromTarget: Target closed.
ERROR:asyncio:Future exception was never retrieved
future: <Future finished exception=NetworkError('Protocol error (Target.sendMessageToTarget): No session with given id')>
pyppeteer.errors.NetworkError: Protocol error (Target.sendMessageToTarget): No session with given id
Page 2: Found 10 reviews
Scraping page 3...
ERROR:asyncio:Future exception was never retrieved
future: <Future finished exception=NetworkError('Protocol error Target.detachFromTarget: Target closed.')>
pyppeteer.errors.NetworkError: Protocol error Target.detachFromTarget: Target closed.
ERROR:asyncio:Future exception was never retrieved
future: <Future finished exception=NetworkError('Protocol error (Target.sendMessageToTarget): No session with given id')>
pyppeteer.errors.NetworkError: Protocol error (Target.sendMessageToTarget): No session with given id
Page 3: Found 10 reviews
Scraping page 4...
ERROR:asyncio:Future exception was never retrieved
future: <Future finished exception=NetworkError('Protocol error Target.detachFromTarget: Target closed.')>
pyppeteer.errors.NetworkError: Protocol error Target.detachFromTarget: Target closed.
ERROR:asyncio:Future exception was never retrieved
future: <Future finished exception=NetworkError('Protocol error (Target.sendMessageToTarget): No session with given id')>
pyppeteer.errors.NetworkError: Protocol error (Target.sendMessageToTarget): No session with given id
Page 4: Found 10 reviews
```

---

2) Data Cleaning & Preprocessing

The raw review text contained punctuation, inconsistent formatting, and irrelevant tokens. A classical text-cleaning pipeline was applied:

Text Cleaning Steps

1) Removal of special characters and HTML fragments

2) Lowercasing and whitespace normalization

3) Stopword removal using NLTK

4) Token filtering (alphabetic tokens only, length > 2)

5) Construction of `cleaned_review`

6) Computation of derived fields:

   a) `review_length` (word count)

   b) `sentiment` (TextBlob polarity)

   c) `sentiment_category` (Positive/Neutral/Negative)

This allowed the dataset to be normalized for further analysis.

Table showing raw vs cleaned review comparison:

```
[16]
✓ 0s
# Display cleaned data info
print(f"\nOriginal dataset shape: {df.shape}")
print(f"Cleaned dataset shape: {df_clean.shape}")
print("\nSample of cleaned reviews:")
print(df_clean[['review', 'cleaned_review']].head(3))

...  Missing values in each column:
name      0
title     0
rating    0
review    0
dtype: int64

Data types:
name       object
title      object
rating    float64
review     object
dtype: object

Original dataset shape: (100, 4)
Cleaned dataset shape: (100, 5)

Sample of cleaned reviews:
                                              review  \
0  Excellent phone. Received today with 1 day del...
1  Nice colorAwesome CameraExcellent DesignSuperb...
2  The budgeted phone and camera quality is aweso...

                                      cleaned_review
0  Excellent phone Received today with 1 day deli...
1  Nice colorAwesome CameraExcellent DesignSuperb...
2  The budgeted phone and camera quality is aweso...
```

---

3) Exploratory Data Analysis (EDA)

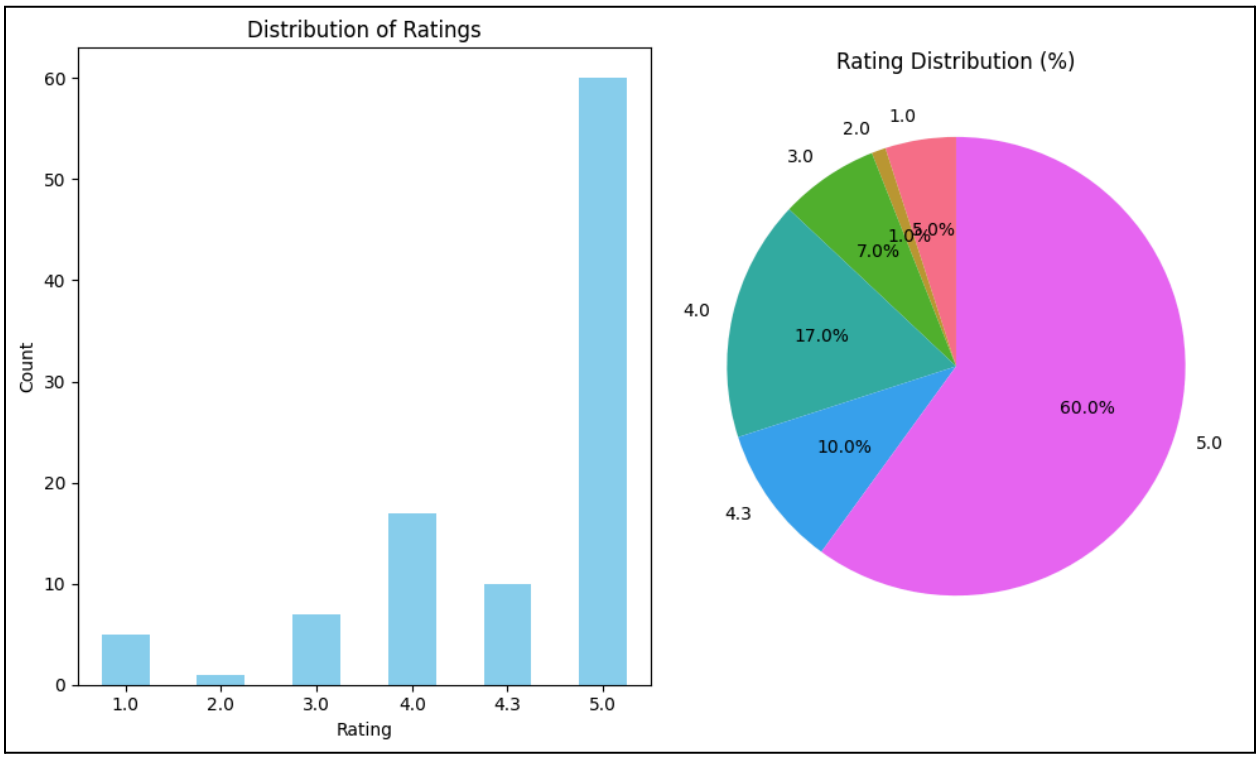
Multiple visual and statistical methods were applied to understand distributional characteristics of the data.

Rating Distribution:

Ratings were analyzed to understand overall product perception.

1. Average rating: 4.39
2. 87% of customers rated 4★ or higher

Rating Distribution Bar Chart & Pie Chart :



Sentiment Analysis

Sentiment scoring was performed using TextBlob (lexicon-based, non-neural method). Each review was categorized as Positive, Neutral, or Negative.

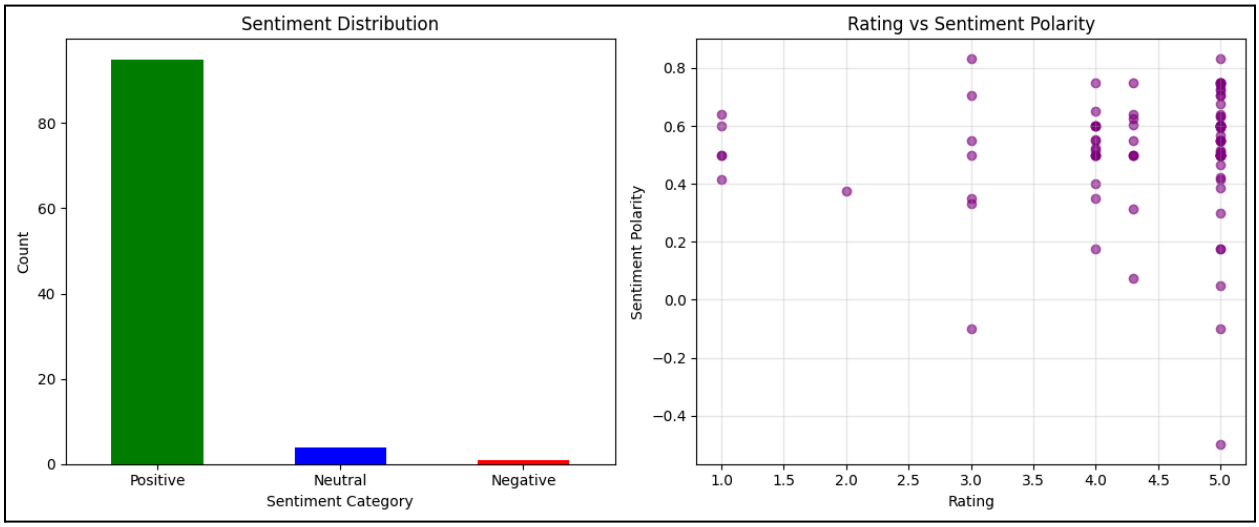
Results:

1. Positive: 95

2. Neutral: 4

3. Negative: 1

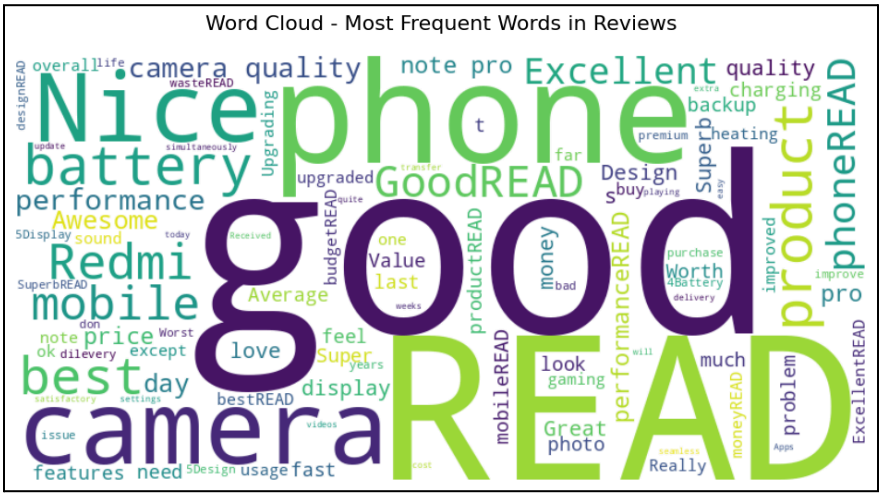
Sentiment Category Bar Chart & Rating vs Sentiment Scatter Plot :



Word Frequency & Word Cloud Visualization

Word frequency analysis identifies commonly used terms such as good, camera, battery, performance, redmi.

Overall Word Cloud :

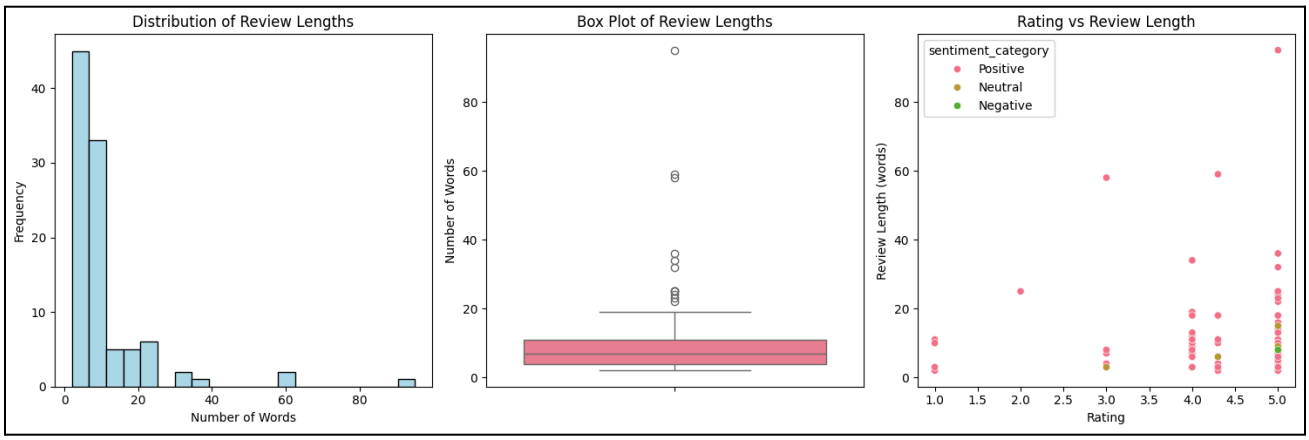


Positive & Negative Reviews Word Cloud :

Review Length Analysis

1. Average review length: 10.8 words
2. No correlation between rating and review length

Histogram, Box Plot & Scatter Plot Review Lengths:



---

4) Feature Mentions (Keyword-Based Analysis)

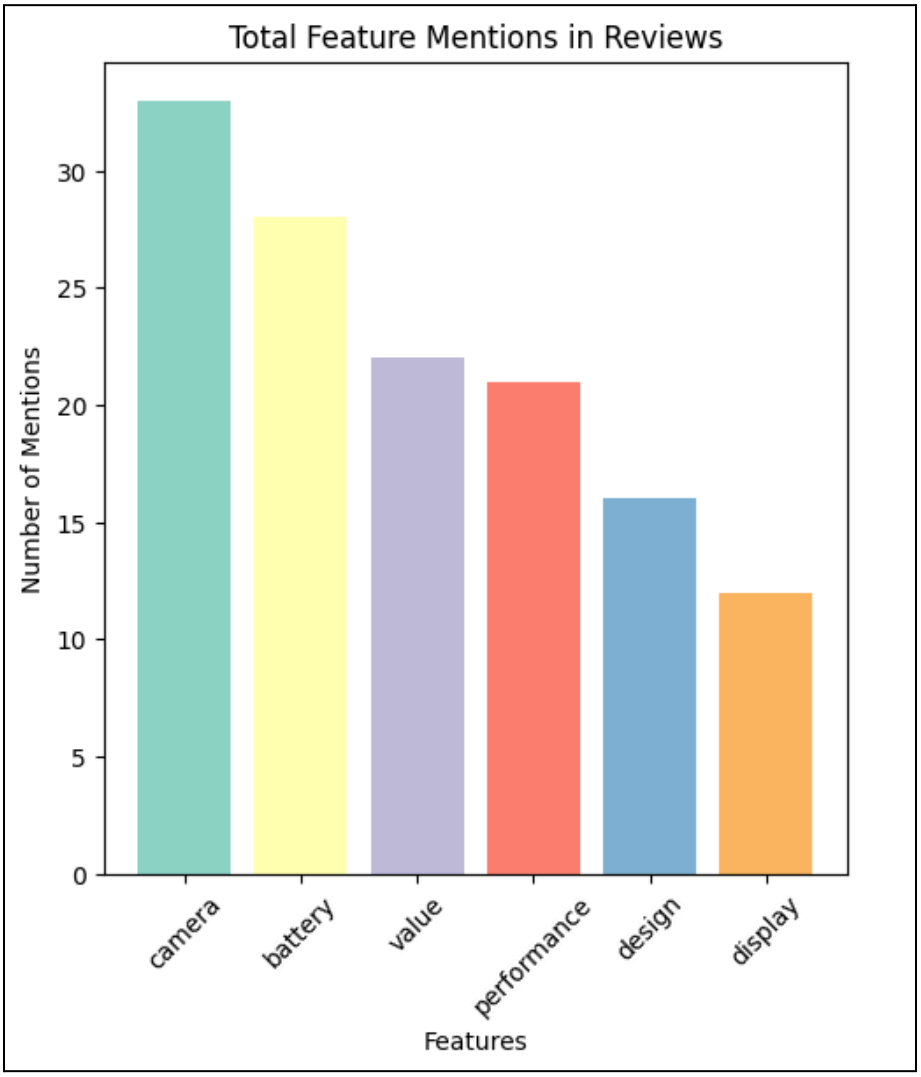
Feature-specific keywords were tracked for six product characteristics: Camera, Battery, Performance, Display, Design, Value for Money

Findings:

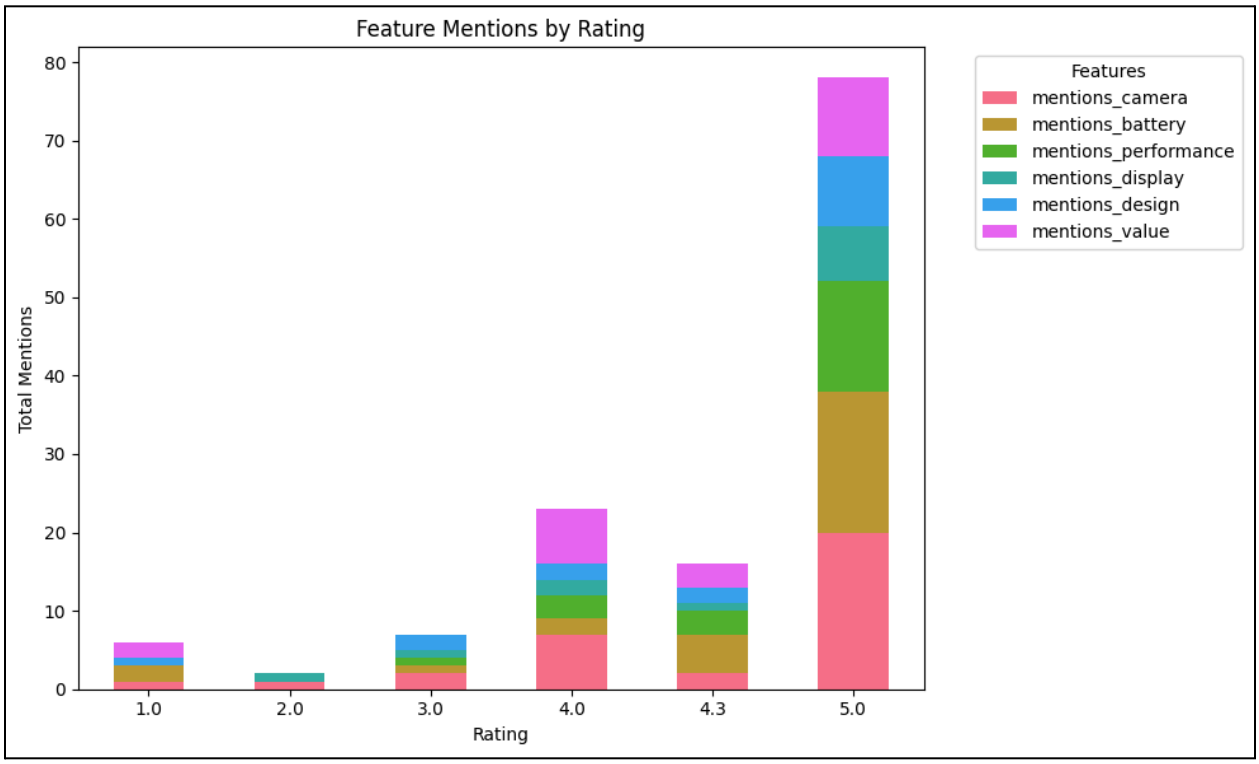
1. **Most mentioned: Camera (33)**

2. **Then: Battery (28), Value (22)**

Feature Mention Bar Chart :



Stacked Feature Mentions by Rating :



---

5) Statistical Tests

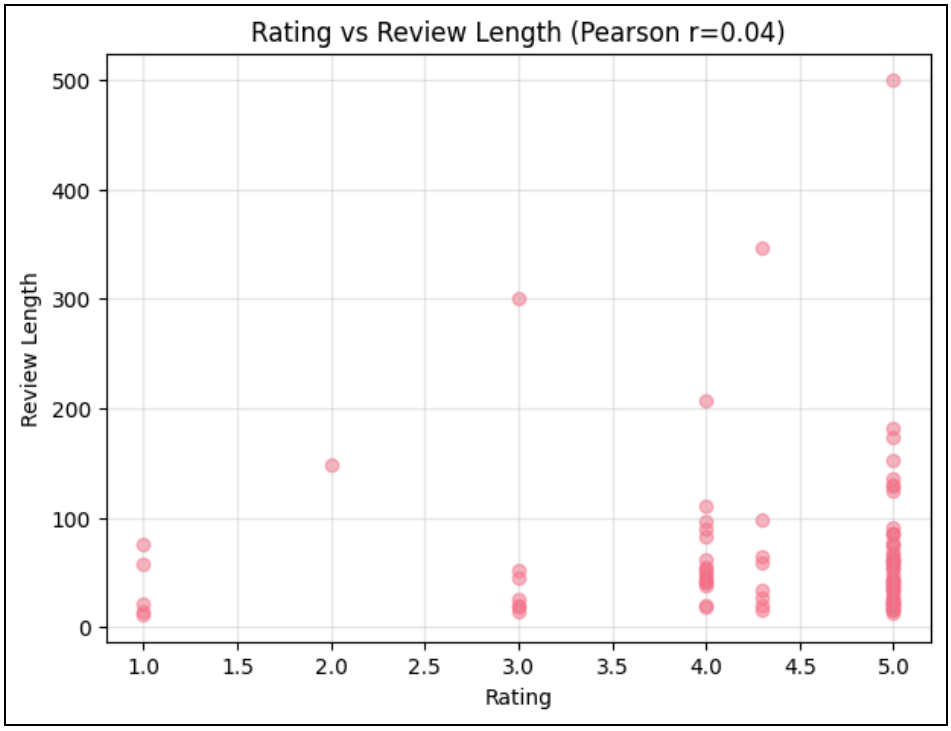
Classical significance tests were performed to validate observed patterns.

Tests Conducted

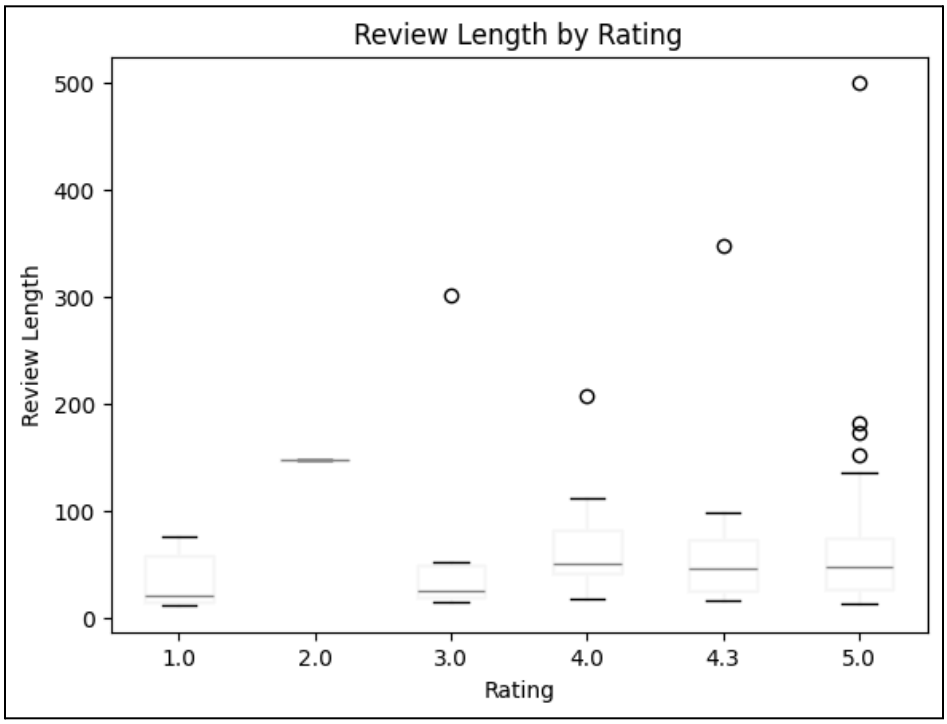
1. Shapiro–Wilk Test for normality of rating distribution
2. Pearson & Spearman Correlation between rating and review length
3. Independent T-test between high-rating (≥4.3) and low-rating (≤3) review lengths

Results: No statistically significant relationship was found, confirming that review length does not affect rating.

Scatter Plot with correlation value:



Boxplot comparing high vs low review length:



Results

1) Comprehensive Analysis Summary

This section provides a consolidated view of all key metrics obtained from the classical NLP analysis performed on the 100 customer reviews of the Redmi Note 14 Pro.

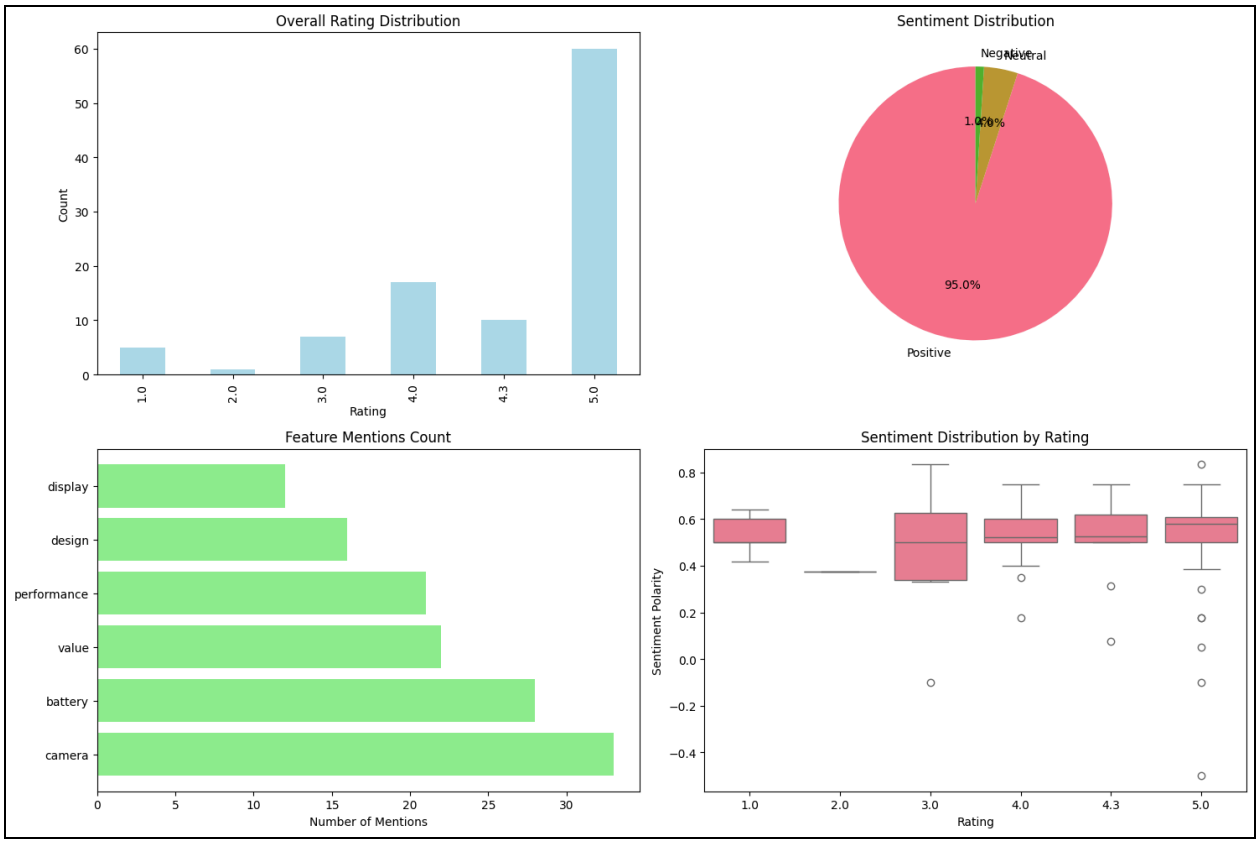
Overall Statistics

1. Total Reviews Analyzed: 100
2. Average Rating: 4.39 / 5
3. Positive Sentiment: 95 reviews
4. Neutral Sentiment: 4 reviews
5. Negative Sentiment: 1 review

Key Findings

1. Most Discussed Features:
    a. Camera: 33 mentions
    b. Battery: 28 mentions
    c. Value for Money: 22 mentions
2. Highest sentiment observed: Rating 5.0 → Avg. polarity 0.532
3. Lowest sentiment observed: Rating 2.0 → Avg. polarity 0.375
4. Overall positivity: ~87% of reviews are 4 star or higher

Rating & sentiment Overview:



---

2) Business Insights & Recommendations

This subsection summarizes actionable insights derived from user sentiment, keyword frequency, and feature-level analysis.

1) Overall Performance

1. Reviews analyzed in this stage: 93 (after filtering)
2. Average Rating: 4.36 / 5
3. Positive Reviews (4–5 stars): 86%

2) Key Strengths (User Mentions)

These keywords frequently appeared in positive reviews:

Keyword	Mentions
Good	**38**
Excellent	**7**
Awesome	**6**
Great	**4**
Best	**11**
Nice	**15**
Fast	**3**
Easy	**1**

These highlight user satisfaction with performance, usability, and product quality.

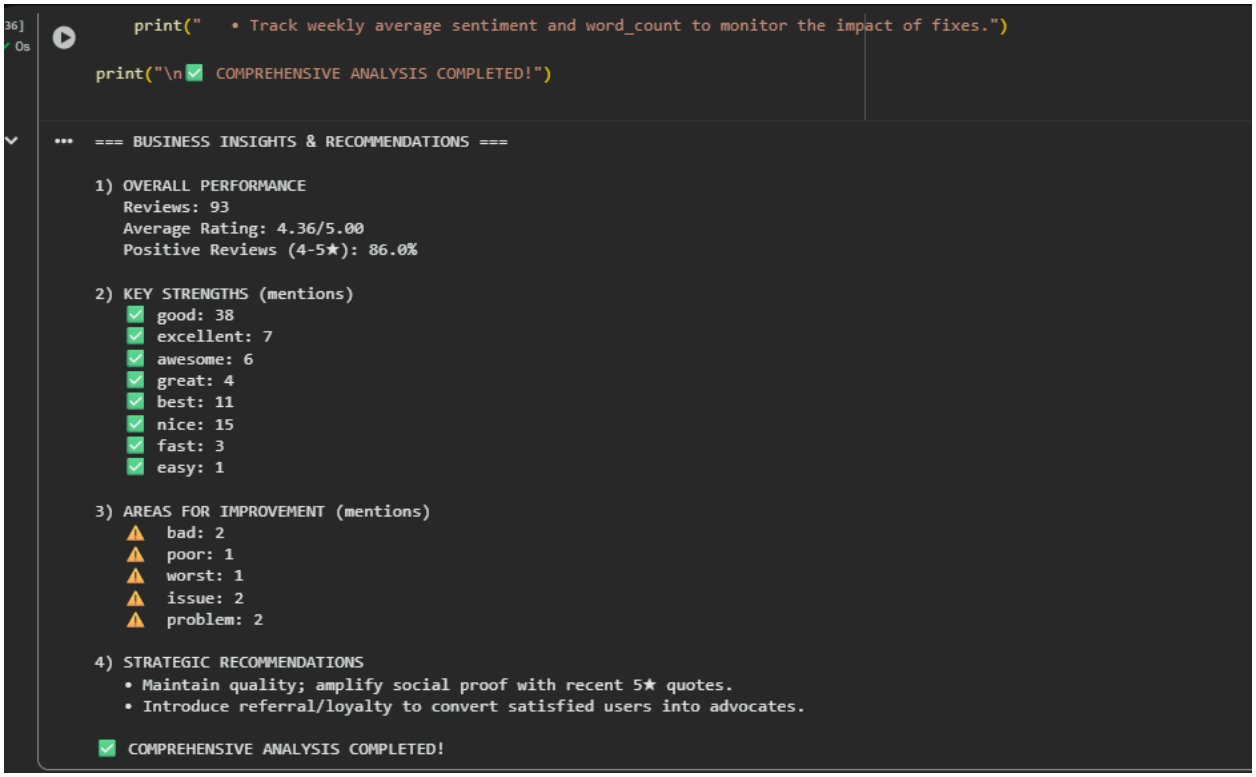
3) Areas for Improvement

Keywords found in negative or neutral contexts:

Keyword	Mentions
bad	**2**
poor	**1**
worst	**1**
issue	**2**
problem	**2**

Although minimal, these indicate potential concerns around product defects or user expectations.

Image :


print("   • Track weekly average sentiment and word_count to monitor the impact of fixes.")
print("\n✅ COMPREHENSIVE ANALYSIS COMPLETED!")
=== BUSINESS INSIGHTS & RECOMMENDATIONS ===
1) OVERALL PERFORMANCE
Reviews: 93
Average Rating: 4.36/5.00
Positive Reviews (4-5★): 86.0%
2) KEY STRENGTHS (mentions)
✅ good: 38
✅ excellent: 7
✅ awesome: 6
✅ great: 4
✅ best: 11
✅ nice: 15
✅ fast: 3
✅ easy: 1
3) AREAS FOR IMPROVEMENT (mentions)
⚠ bad: 2
⚠ poor: 1
⚠ worst: 1
⚠ issue: 2
⚠ problem: 2
4) STRATEGIC RECOMMENDATIONS
• Maintain quality; amplify social proof with recent 5★ quotes.
• Introduce referral/loyalty to convert satisfied users into advocates.
✅ COMPREHENSIVE ANALYSIS COMPLETED!

Challenges & Learnings

1) Challenges

1. Scraping Blockages: Flipkart frequently blocked requests, causing session drops, “Target closed” errors, and failed page loads.

2. Dynamic Content: Heavy JavaScript made basic requests and Selenium unreliable; required pyppeteer for proper rendering.

3. Text Irregularities: Reviews contained merged words, emojis, inconsistent casing, and short sentences, requiring careful preprocessing.

4. Short Review Sentiment Issues: Lexicon-based models struggle with single-word reviews like “Good” or “Nice.”

5. Feature Keyword Coverage: Users described features in many forms, making it challenging to build complete keyword sets.

---

2) Learnings

1. Asynchronous scraping with pyppeteer is much more effective for JS-heavy websites.

2. Classical NLP techniques (stopwords, word frequency, sentiment polarity) still provide strong, interpretable insights.

3. Good preprocessing (cleaning, normalization) significantly improves downstream analysis.

4. Visualizations make patterns easy to understand and validate.

5. Feature-level signals such as Camera, Battery, and Performance strongly reflect customer satisfaction.

Conclusion

This analysis of 100 customer reviews for the Redmi Note 14 Pro demonstrates that the product enjoys a strong overall reception, with an average rating of 4.39 and nearly 87% of users giving 4star or higher. Classical NLP techniques—such as text cleaning, sentiment scoring, keyword frequency analysis, and statistical testing—proved effective in extracting meaningful insights from unstructured review data.

Customers consistently praised the camera quality, battery performance, and overall value for money, which appeared as the most frequently mentioned features. Sentiment analysis further confirmed overwhelmingly positive user experience, with only a minimal number of negative reviews. Statistical tests indicated no significant relationship between review length and rating, supporting the stability of observed patterns.

Overall, the Redmi Note 14 Pro is viewed highly favorably by users, and the analysis highlights the importance of maintaining camera and battery performance as key differentiators. This classical NLP pipeline can be extended to monitor future product releases, benchmark competing devices, or automate large-scale review analytics.